POLITECNICO DI TORINO
Repository ISTITUZIONALE

Applicability of human-robot collaboration to small batch production

*Original*
Applicability of human-robot collaboration to small batch production / Antonelli, D., Astanin, S., Bruno, G.. - STAMPA. - 480:(2016), pp. 24-32. (17th IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2016 Porto (Portogallo) 2016) [10.1007/978-3-319-45390-3_3].

*Availability:*
This version is available at: 11583/2654340 since: 2016-10-26T17:10:49Z

*Publisher:*
Springer New York LLC

*Published*
DOI:10.1007/978-3-319-45390-3_3

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

13 October 2026

# Applicability of human-robot collaboration to small batch production

Dario Antonelli, Sergey Astanin, Giulia Bruno

Politecnico di Torino, Department of Management and Production Engineering,
Corso Duca degli Abruzzi 24, 10129 Torino, Italy
{dario.antonelli, sergey.astanin, giulia.bruno}@polito.it

**Abstract.** In the past years, the use of industrial robots was on the rise. Today, they are used for mass production in large enterprises. On the contrary, deployment of robots in small enterprises is lagging behind. Programming time is usually two orders of magnitude larger than the cycle time, and this fact severely limits applicability of robots in small batch production. This work analyzes opportunities to profitably employ robots also in this case through the adoption of the human-robot collaboration. A collaboration paradigm is proposed, where the tasks are assigned to robotic and human workers based on the batch size, task programming complexity (time), and manual execution time. The validity of this approach is demonstrated in a case study conducted in a collaborative human-robot work cell.

**Keywords:** Human-robot collaboration, collective ability, manufacturing cell, small batch production, robotics.

## 1 Introduction

For small production volume, some manufacturing operations (e.g., machine loading and unloading, part inspection, part cleaning, bin picking, kitting) are largely done manually [1]. In contrast, a lot of these operations are performed by robots in mass production lines, such as in the automotive industry [2,3]. This fact clearly shows the potential of robots for the execution of certain manufacturing operations.

Significant efforts are needed to overcome the barriers that prevent SMEs from automating their processes, since many of the traditional industrial robots are not considered useful in this context due to high setup and maintenance costs, safety concerns, and lack of general-purpose capabilities [4]. Setting up purely robotic work cells is not a viable option for most SMEs, because the economic efficiency of robots' use is often undermined by (i) the costs of the implementation of a fully automatic work cell and (ii) the costs and the duration of robot programming for every new task.

Human-robot collaboration can address these issues by relying on human workers for the tasks which are expensive to automate. There are two kinds of human-robot collaboration: spatial collaboration (actors share the workspace but don't work simultaneously) and temporal collaboration (actors work simultaneously but in different spaces) [5]. Spatial collaboration is the easiest to implement using the off-the-shelf industrial robots, so this is the kind of collaboration this work refers to.

In this paper the issue of assessing the economic advantage of a collaborative human-robotic cell is addressed from the point of view of SMEs. They have to evaluate if the costs of setup and maintenance are affordable even for small batch sizes. Due to the fact that humans and robots share complementary strengths in performing the tasks commonly arising in small volume production, the idea is to decompose manufacturing operations into such tasks that humans and robots can perform those which are most suitable for each of them.

This allows to offset the greater part of the programming time, and it can be a viable strategy in small batch production. The validity of this approach is demonstrated in a case study conducted in a collaborative human-robot work cell. The comparison between the total time in a collaborative scenario and the total time in a fully manual production can be used as a criterion to evaluate the applicability of the human-robot collaboration for a specific batch size and task structure.

The rest of the paper is organized as follows. Section 2 revises the state of the art in literature. Section 3 describes the proposed human-robot collaboration paradigm and compares it to fully automatic and fully manual production scenarios. In Section 4, the validity of the proposed approach was shown in a collaborative human-robot work cell. Finally, Section 5 draws conclusions and states future works.

## 2 State of the art

Collaboration between human and robotic workers is gaining traction in manufacturing [6]. Several works addressed the evaluation of collaborative robotic cells, especially in automotive industry, providing comparisons between conventional robotic cell and cooperating human-robot cell [7,8,9]. Optimal task assignment among workers and robots was studied in some use-cases [10]. Major efforts have also been devoted to the safety of human-robot collaboration [11,12].

The programming time can be reduced by automatic inference of robot programs from observations of a human teacher or from executions controlled by a human. These methods are known as Programming by Demonstration (PbD) [13] or Learning by Demonstration [14]. Early PbD implementations, still dominant in the field of industrial robotics, were explicitly communicating the desired robot configurations [15]. However the configuration can often be inferred. It opens the way to goal-oriented imitation learning and programming of more complex motions [16]. A goal-oriented PbD system for a welding robot, used in this study, was presented in [4, 17].

Collaborative robotic cells are ripe to move from labs into the bigger world. Several approaches to facilitate robot programming exist. What's missing is a solid methodology to evaluate applicability of such cells. Criteria for high-level task partitioning to divide tasks between a human and a robot worker, have to be defined.

## 3 Human-robot collaboration paradigm

Both humans and robots have some advantages regarding the specific tasks they can accomplish. While robots are superior at repetitive and simple tasks, humans are

better suited for variable and complex operations, which are hard to be programmed in advance. Taking advantage of the both actors by means of human-robot collaboration may reduce the total production time even in small batch production.

We divided all manufacturing tasks into two kinds: the tasks which are easy to program and the ones whose programming is complicated. The time of programming is approximately proportional to the number of key points of the robot trajectory that have to be recorded, thus the programming time of the easily programmable tasks is much shorter than the time required for complicated tasks.

In the same time, the complexity of the programming does not reflect the complexity of the manual execution. Sometimes, for humans, executing tasks that are easily to program can be difficult, while executing tasks that are complicated-programmable can be easier (e.g., for a human worker it is more difficult to perform a welding along a straight line than along a curved line). By taking into account these considerations, it is possible to find an optimal assignment of tasks to robotic and human workers.

Let's consider three different scenarios of process execution: (i) fully automatic, where all tasks are entirely executed by a robot, (ii) fully manual, where all tasks are executed by a human worker, and (iii) collaborative, where some tasks are executed by a robot and some task by a human. Fig.1 graphically shows the differences among programming time and execution times for the three scenarios.

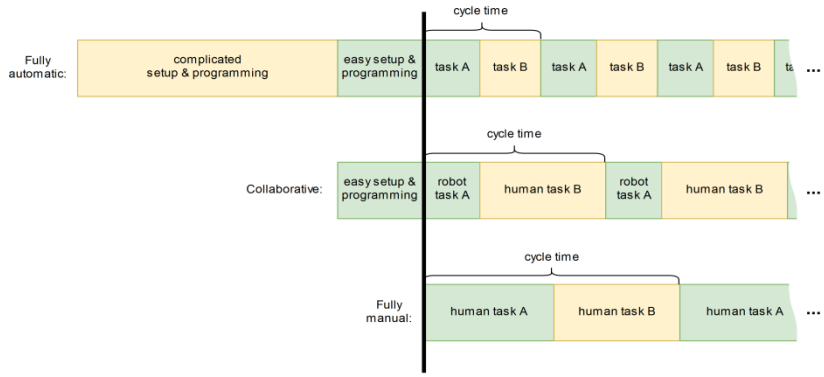



**Fig. 1.** Comparison of programming and working times for the three considered scenarios: fully automated, collaborative and fully manual.

In the fully automatic scenario, both easily programmable tasks (A) and complicated programming tasks (B) are executed by a robot. On the one hand, this implies a long programming time required before the batch execution can begin. On the other hand, we may expect that task execution times are as short as possible, so the cycle time is minimized. Fully automated execution also allows eliminating non-value adding operations, which are unavoidable in manual production. This scenario is the most efficient for large batch sizes, where the programming time is negligible with respect to the batch execution time.

In the fully manual scenario, no programming is required, but the cycle time is higher than for the robot. Thus, in small batch production, this scenario can be more efficient, because there is no time spent up-front.

The collaborative scenario exploits the paradigm of assigning easily programmable tasks to the robot and the other tasks to the human operator. In this way, the programming time is noticeably reduced with respect to the fully automated scenario. While the cycle time is increased with respect to the fully automatic procedure, it should still be shorter than in the fully manual scenario. This scenario is the most efficient for intermediate batch sizes, depending on the programming and cycle time.

In this discussion we have implicitly assumed that the setup and programming times are dominated by programming. However, even in scenarios where setup time is not negligible, setup tends to be more complex for fully automated tasks because it may require designing and producing auxiliary equipment and developing new automation procedures. For the sake of simplicity, this paper is focused on the case where the programming time dominates and the setup is trivial.

In the following we estimate the batch size ($N$) for which the collaborative scenario is the most efficient. The total time to produce $N$ items is the sum of the programming time $T_{prog}$ and the batch execution time $NT_{cycle}$:

$$T_{total} = K \cdot T_{prog} + N \cdot T_{cycle}. \quad (1)$$

In order to take into account the necessity to make debugging runs and to repeat the entire programming process until the desired result is achieved, a debugging coefficient $K$ is introduced to indicate how many times the programming phase is longer than the time required to just record the trajectory.

The programming time depends by three factors: the time required to code the easily programmable tasks ($T_{prog_E}$), the time required to code the complicated programming tasks ($T_{prog_C}$), and programming preparations time ($T_{prep}$) required to prepare and return the robot, position the end effector, leave the cell and disable the programming mode:

$$T_{prog} = T_{prep} + T_{prog_E} + T_{prog_C}. \quad (2)$$

The cycle time is the sum of for each task belonging to the working sequence of two times [18]: the operative time needed to perform the manufacturing process ($T_{op}$), and the non-value adding time spent for set-up, item loading and unloading, tool maintenance and tool positioning ($T_{nva}$):

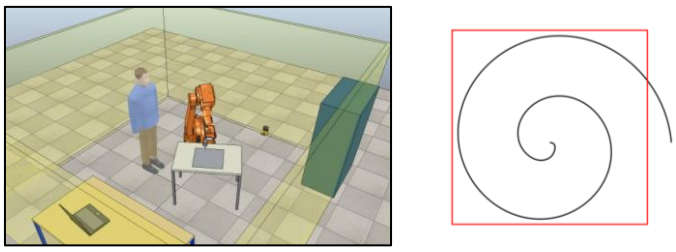
$$T_{cycle} = \sum_{i=1}^{\#Tasks} \left(T_{op}(i) + T_{nva}(i)\right) \quad (3)$$

Our aim is to compute the total time $T_{total}$ for the three different scenarios described above (fully automatic, fully manual and collaborative) to determine the range of batch sizes for which the collaborative scenario is the most efficient.

## 4 Case study

The case study to prove the validity of the proposed approach was conducted in a collaborative human-robot work cell, which consists of a COMAU NS16 Robot, two laser scanners for human-robot collision avoidance, and several IR cameras for demonstration observation. The layout of the cell is shown in Fig. 2(a), and more details can be found in [4]. Safety issues have been completely avoided in this setup as only the programming phase was studied, and the operator was removed from the cell before test runs. The benchmark process model considered in the case study is composed by two tasks: (1) a welding task along a four straight line segments (a square) and (2) a welding task along an Archimedean spiral. The geometry of both paths are depicted in Fig. 2(b), and they have the same length of 643 mm.

The number of points, which the programmer has to indicate, is different for two trajectories. 5 points are enough to define the closed square trajectory. As the spiral doesn't have any straight line or circle arc segments, it is approximated by a polyline with 70 points on the curve. The square path is considered to be an easily programmable task (task A), conversely the spiral is considered to be the complicated one (task B).



(a) (b)

**Fig. 2.** Layout of the collaborative human-robot cell (a), and benchmark trajectories to be programmed and executed (b).

In this benchmark, the following programming times are considered:

- the programming preparations time is the time required to switch the robot to programming mode, enter the robot cell, set appropriate robot movement parameters, position the end effector in the proximity of the work area, and return the robot to the starting position after the programming is done, leave the cell and disable the programming mode;
- the time for easily programmable tasks is the time needed to program the rectilinear path segments, which are defined by few distant points;
- the time for complicated programmable tasks is the time for programming the curvilinear paths, which may require to put many close points.

As robot programming is a creative, not a repetitive, activity, references and standard time tables for robot programming don't exist, to the extent of our knowledge. So the programming time had to be measured experimentally. We

established that in our case the programming preparations time $T_{prep}$ is 155±20 seconds, programming of the straight lines requires 41.6±5.6 seconds per point, and programming of the curvilinear path requires 13.3±2.2 seconds per point. Thus the average programming time of the task A is estimated as $T_{prog_E} \approx 170 \pm 11$ s, and the programming time of the task B is $T_{prog_C} \approx 920 \pm 18$ s.

For the sake of simplicity, in this model we assume that the full length of the programming phase is $K$ times longer than the time required to just record the trajectory. In the following we have assumed that $K \approx 10$.

The welding rate is assumed equal to 350 mm/min. This value is the welding rate for a wild steel tick from 1 to 2.5 mm, as reported by a producer of welding machines (www.fimer.com). This value can be substituted with the real rate of the manufacturing process under consideration. The welding rate is independent from the trajectory, thus both tasks have an operative time ($T_{op}$) of 110 s.

The non-value adding times are the set-up, the loading, the unloading, the tool maintenance, and the tool positioning. For human workers, the non-operative times ($T_{nva}$) are usually estimated as 80% of the whole process, while for a robot, they are estimated as 30% of the whole process. Thus, for both tasks, the non-value adding times are set 440 s when the tasks are executed by a human worker, and 50 s when the tasks are executed by a robot.

All the computed times are reported in Table 1, for the three considered scenarios. The total programming time is 1250 s for the fully automatic scenario and 330 s for the collaborative scenario. No programming time is required for the fully manual scenario. The cycle time is 320 s for the fully automated scenario, 1100 s for the fully manual, and 710 s for the collaborative one.

**Table 1.** Operation times in fully automated, fully manual and collaborative scenarios.

| Times | Fully automated [s] | Fully manual [s] | Collaborative [s] |
|---|---|---|---|
| $T_{prep}$ | 160 | 0 | 160 |
| $T_{prog\ E}$ | 170 | 0 | 170 |
| $T_{prog\ C}$ | 920 | 0 | 0 |
| $\boldsymbol{T_{prog}}$ | **1250** | **0** | **330** |
| $T_{op}(taskA)$ | 110 | 110 | 110 |
| $T_{nva}(taskA)$ | 50 | 440 | 50 |
| $T_{op}(taskB)$ | 110 | 110 | 110 |
| $T_{nva}(taskB)$ | 50 | 440 | 440 |
| $\boldsymbol{T_{cycle}}$ | **320** | **1100** | **710** |

By applying (1), the total time depending on the batch size ($N$) was computed in the three scenarios, and the results are shown in Fig.3. As expected, for very small batch sizes (from 1 to 8), the fully manual scenario is the more efficient due to the fact that no programming time is needed. For large batch size ($N$ greater than 26), the fully automated scenario is the best, because the programming time becomes

negligible with respect to the total cycle time. Collaborative approach is the most efficient for the batch size in the range between 9 and 25 units.

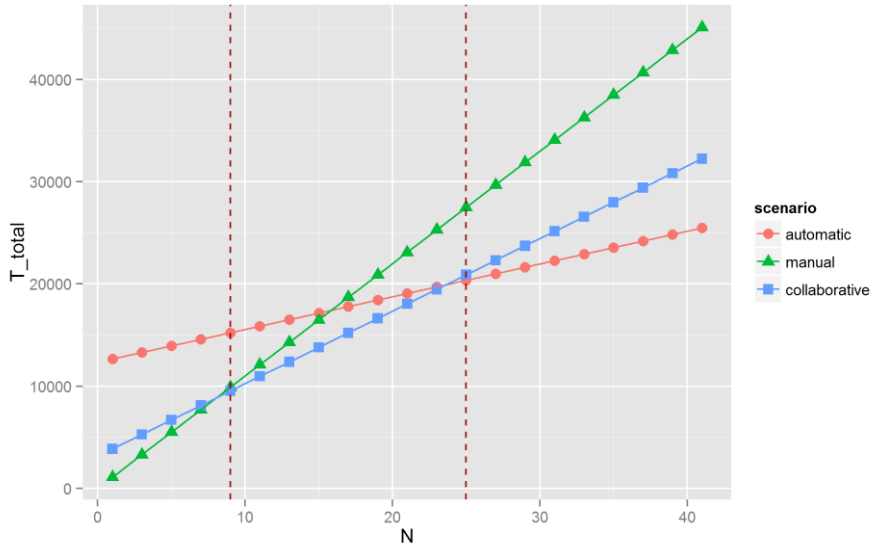



**Fig. 3.** Comparison of the total times in the three scenarios showing the range of the batch size in which the collaborative scenario is the most efficient. $K = 10$.

Notably, the range of batch sizes when collaborative scenario is the most efficient can be increased if the programming time of the task A can be further reduced. One of the possible approaches to reduce programming time even further is to employ programming by demonstration techniques proposed in [4]. Reduction of the programming time would shift down the line of the collaborative total cost in Fig. 3, thus increasing the efficient range of the collaborative scenario. However even users of the conventional programming approach can often benefit from collaboration as clearly shown above.

## 5 Conclusion

In this work, we propose a collaboration paradigm for assigning easily programmable tasks to the robot and the other tasks to the human operator. Robot programming time can prevail over other operations in small batch production, which is often the reason to avoid using robots altogether. We propose that eliminating only part of this time can open opportunities to employ robots in small batch production. The programming times of different tasks of an artificial use case were found experimentally in a real robotic cell. A basic model allowed to estimate the range of batch sizes for which the collaborative scenario was the most efficient.

Future works will consider the impact of rapid programming by demonstration techniques in order to further reduce the programming time and improve viability of collaborative scenarios.

## References


1. Banerjee, A.G., Barnes, A., Kaipa, K.N., Liu, J., Shriyam, S., Shah, N., Gupta, S.K. An Ontology to Enable Optimized Task Partitioning in Human-Robot Collaboration for Warehouse Kitting Operations, Next-Generation Robotics II and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX (2015)
2. Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., Chryssolouris, G. Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach, CIRP J. Manuf. Sci. Technol., 2 (2), pp. 81–91 (2010)
3. Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., Fraisse, P. Collaborative manufacturing with physical human–robot interaction, Robotics and Computer-Integrated Manufacturing, 40, 1–13 (2016)
4. Antonelli, D., Astanin, S., Caporaletti, G., Donati F. FREE: Flexible and Safe Interactive Human-Robot Environment for Small Batch Exacting Applications. Gearing Accel. Cross-Fertil. Acad. Ind. Robot. Res. Eur., Springer, pp. 47–62 (2014)
5. Bicchi, A., Peshkin, M.A., Colgate, J.E. Safety for physical human–robot interaction, Springer handbook of robotics, Springer Berlin Heidelberg, pp. 1335-1348 (2008)
6. Tan, J. T. C., Duan, F., Zhang, Y., Watanabe, K., Kato, R., Arai, T. Human-robot collaboration in cellular manufacturing: Design and development, in: IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp. 29-34 (2009)
7. Morioka, M., Sakakibara, S. A new cell production assembly system with human–robot cooperation, CIRP J. Manuf. Sci. Technol., 59, pp. 9–12 (2010)
8. Papakostas, N., Michalos, G., Makris, S., Zouzias, D., Chryssolouris, G. Industrial applications with cooperating robots for the flexible assembly, Int. J. Comput. Integr. Manuf., 24 (7), pp. 650–660 (2011)
9. Pedrocchi, N., Vicentini, F., Malosio, M., Tosatti, L.M. Safe human–robot cooperation in an industrial environment, Int. J. Adv. Robot. Syst., IJARS, 10 (27) (2012)
10. Ding, H., Schipper, M., Bjoern, M. Optimized task distribution for industrial assembly in mixed human–robot environments – case study on IO module assembly, IEEE International Conference on Automation Science and Engineering (2014)
11. Harper, C., Virk, G. Towards the development of international safety standards for human robot interaction. Int. J. Social Robotics, 2(3), pp. 229-234 (2010)
12. Matthias, B. et al. Safety of collaborative industrial robots: Certification possibilities for a collaborative assembly robot concept. IEEE ISAM. (2011)
13. Billard, A., Calinon, S., Dillmann, R., Schaal, S. Robot programming by demonstration. In Springer handbook of robotics, Springer Berlin Heidelberg, pp. 1371-1394 (2008)
14. Argall, B.D., Chernova, S., Veloso, M., Browning, B. A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5), 469–483 (2009)
15. Biggs, G., MacDonald, B. A survey of robot programming systems. In Proceedings of the Australasian conference on robotics and automation, pp. 1-3 (2003)
16. Schaal, S., Atkeson, C. G. Learning control in robotics. Robotics & Automation Magazine, IEEE, 17(2), pp. 20-29 (2010)
17. Antonelli, D., Astanin, S. Qualification of a collaborative human-robot welding cell, Procedia CIRP, 41, 352 – 357 (2016)
18. Hopp, W.J., Spearman, M.L., Factory Physics (2011)